VBD User Manual

G. Briskin[1], D. Cutts[2], G. Watts[3] and R. Zeller.

Abstract.

This manual will briefly describe the operations of VBD and explain in detail the parameters of the control memory that need to be programmed for VBD operation. This manual is intended as an introduction and as an expert's reference for Run 2 users. The manual reflects updates to the VBD since Run 1.

[1] biskin@hep.brown.edu
[2] cutts@hep.brown.edu
[3] gwatts@fnal.gov

Introduction

In the first part of this manual a programming of the *control memory* will be described and examples given. After reading this manual a user should be able to physically configure and program VBD card for normal or diagnostic operation. In the second part of this manual a brief description of the VBD card physical design and principal operation will be given. If you are unfamiliar with the VBD board, you should read the second part first.

VBD Control Memory Programming

Control Memory:

The following are the offsets from the base of the control memory region (the base address is selected by jumpers CJ1, see VBD physical design and operation section) for the various soft configuration parameters. All values except data pointers (data starting address locations) are 16 bit; the DATAPOINTERS are 32 bit values.

Only word addressing, short I/O (A16:D16), of the control memory is supported. To program 32 bit DATAPOINTER address, two word transfers are required.

Note: In the original definition of how the VBD should work, it was decided that "data" would be in an A24 address space, and word counts, event number, crate id and control memory access would be in A16 space. The "data" MUST be D32 enabled (i.e. all data blocks must start on a long-word boundary), but could also be accessible via D16 or D8 at the user discretion. The control, word count, etc… registers MUST be D16 accessible, and again could be accessed via D8 at the user discretion.

The control memory registers offsets are (*control memory is battery backed up*):

Value	***Offset***
CSR0	0x0
CSR1	0x2
CSR2	0x4
CSR3	0x6
Crate Type	0x8
Event Number	0xA
Crate ID	0xC
D-Control	0x10
P-Control	0x12
I/O Address	0x14
ERRCNT	0x100
ERRORLIST	0x104
WCPOINTER	0x1000
DATAPOINTER	0x1800

To summarize the VBD according to the original specs was designed to operate as follows:

- in reading word counts, Crate ID and Event # -- A16:D16
- in reading data (DMA) -- A24:D32

Other modes of operation are available with the proper parameter value settings in the control-memory and are described in greater details below.

Note: The VBD reads the control memory parameters (i.e. D-Control, P-Control, DATAPOINTERS,…) into the PAC memory on power up or reset. If you change any of the control memory parameters do not forget to reset the VBD.

CSR's:

The CSR's provide control of the VME and external port controllers. When writing into CSR all of the bits must be specified. For individual bit changes, read-modify-write instructions should be used. Some bits are specified as write-only (e.g. reset), and read back as 0. The error bit is read once, i.e. it is cleared upon reading (destructive).

CSR0 is used for software control of the VBD operations. This register is not used during normal (DAQ) operation. This register, in normal running, should only be used during control memory modification, bit <4>. CSR0 bit definitions are as follows:

Bit number	Description
<15>	Error on CSR write
<14-8>	Reserved
<7>	Reset VBD
<6>	Reset error FIFO
<5>	0 = enable transmit VBD buffer following DMA 1 = disable transmit VBD buffer following DMA
<4>	0 = enable writing into control memory 1 = disable writing into control memory
<3-2>	Select DMA time-out value
<1>	DMA buffer select (0 = VBD Buffer 0, 1 = VBD Buffer 1)
<0>	DMAGO

Bit <0> -- Writing 1 into this location will initiate the transfer of data from the preprogrammed starting address locations (i.e. using information stored in WCPOINTER and DATAPOINTER list) in the VBD control memory.
Bit <1> -- Use this bit to select the VBD buffer into which data will be stored.
Bit <5> -- If this bit is set to zero (0), then upon completion of the data transfer selected VBD buffer control will be switched to the external port. The data stored in the VBD buffer will be dumped onto the external cable once the token or GRANT signal is received. If this bit is set to one (1), then upon completion of the data transfer the VBD buffer will remain on the VME and the data content can be examined. This later mode is intended for the hardware debugging purposes.

CSR1 is used to control the external port. *Note: Not used for routine operation*.

CSR2 is used to report board status as follows (all error flags are cleared on read):

Bit number	Description
<15>	DMA error
<14>	Read-out error
<13>	Token error
<3>	DMA in progress, buffer 1
<2>	DMA in progress, buffer 0
<1>	Buffer 1 status (0 = VME, 1 = external port)
<0>	Buffer 0 status (0 = VME, 1 = external port)

CSR3 is reserved for use by ZRL.

Parameter definitions:

(1) **Crate Type**: specifies the mode of operation. Possible values are:

- Calorimeter mode = **0x1**
  The calorimeter data word count values are placed in the VBD buffer prior to the start of the corresponding data block transfer. Zero word count values are still posted to the VBD buffer, even though no corresponding data block transfer is performed, i.e. *<WC0>+<DATA0>+…*

- Central Detector mode = **0x2**
  In the case of the CD mode the word count is imbedded in the data, therefore, the controller does not prepped it, i.e. *<DATA0>+<DATA1>+…*

(2) **Event Number**: pointer to a location of the event number ("L3 Transfer Number") field (16 bits).

(3) **Crate ID**: pointer to the location of the Crate ID (16 bits).

(4) **D-Control**: controls access mode of data and word count reads. Must be programmed to the correct format for the crate environment (16-bit value). Possible values are:

VME Access	Value
A24:D24	0xFD
A16:D16	0xED
A24:D32	0xBD
A32:D32	0x8D

The D-Control 16 bit value is organized as follows <byte0|byte1>:
Where the upper byte (byte0) is for the data access, and lower byte (byte1) is for the word count access. As was said before, the original specification dictated parameter value access in A16 space, therefore to access the parameter values in A24 or A32 space the upper 16 bits need to be obtained. For word count the upper 16 bits are obtained from the upper 16 bits of the DATAPOINTERS values.

For example the **WCPOINTER** for A24 or A32 access is build as follows:
WCPOINTER = **DATAPOINTER** (upper 16 bits) + **WCPOINTER** (lower 16 bits)

(5) **P-Control**: controls access mode of Event Number and crate ID reads. Must be programmed to the correct format for the crate environment (16-bit value). Possible values are:

VME Access	Value
A24:D24	0xFD
A16:D16	0xED
A24:D32	0xBD
A32:D32	0x8D

The P-Control 16 bit value is organized as follows <byte0|byte1>:
Where the upper byte (byte0) is for the Event Number and lower byte (byte1) is for the Crate ID access. As was said before, the original specification dictated parameter value access in A16 space, therefore to access the parameter values in A24 or A32 space the upper 16 bits need to be obtained. For crate ID and event number access the upper 16 bits are obtained from the I/O Address control parameter.

Event Number Address = **I/O Address** (upper 16 bits) + **Event Number** Address (lower 16 bits)
Crate ID Address = **I/O Address** (upper 16 bits) + **Crate ID** Address (lower 16 bits).

Note: If accessing the Event Number in D32 space the Event Number should be put into the upper 2 bytes of a long-word ((EVT_NUM << 16) & 0xFFFF0000) at the Event Number Address location. This is not the case for the Crate ID.

(6) **I/O Address**: Specifies value of upper address bits (A31-A16) during parameter value reads. See P-Control description above.

(7) **ERRCNT**: Count of number of errors accumulated (mod 65536) since error FIFO last reset (see bit 6 in CSR0). Used as pointer into the FIFO region. Cleared on reset of FIFO.

(8) **ERRORLIST**: FIFO list of error conditions (512 entries). The 16 bit error code is organized as follows <byte0|byte1>:
byte0 is the event # and byte1 is the error code.
The error codes are,
1 – stale data
2 – buffer refused by external port
3 – content of memory crate type not set
4 – timeout reading trailer info
5 – timeout reading WC or DATA

(9) **WCPOINTER**: Null terminated list of pointers to word count locations (16 bits).

(10) **DATAPOINTERS**: List of base addresses of data blocks. Long-word values (32 bits).

How to Setup the VBD for different modes of operation.

Before VBD can be put into the front-end crate its control parameter values must be programmed for the correct crate environment. In this section a detail description of how to program the VBD control memory is explained in greater detail.

(1) Allow control memory modification by setting bit <4> in CSR0 to 0.
(2) Set crate type environment by setting CRATE TYPE to one of the appropriate values at offset 0x8
(3) Set Event # Address by writing the location address into control memory at offset 0xA.
(4) Set Crate ID Address by writing the location address into control memory at offset 0xC.
(5) Set data and word count access mode by writing D-Control value into control memory at offset 0x10
(6) Set Event # and Crate ID access mode by writing P-Control value into control memory at offset 0x12
(7) Set parameter access I/O Address modifier 16 bits if your Event # Address and Crate ID address are in A24 or A32 address space. Set them to 0x0000 if the parameters are in A16 space. This is done by writing a value into control memory at offset 0x14
(8) Set the NULL (0) terminated word count list by writing into control memory at offset 0x1000. Each word count address (16 bit value) in a list should point to a location in I/O space where the long-word count (32 bits) number will be present when DMA is issued.
(9) Set the data block pointer list by writing into control memory at offset 0x1800. Each data block pointer is the 32 bit value. They are set by writing 16 bit values at a time, since control memory can only be accessed in A16:D16 mode.
(10) Disallow control memory modification by setting bit <4> in CSR0 to 1.
(11) Issue a reset to the VBD by writing 1 into CSR0 bit <7>. This operation is needed to allow internal PACs to reread the control memory and update internal parameters. After reset allow for 200 msec delay, since the VBD will perform self-diagnostics and will read control parameter values.

Now either software or hardware can trigger a DMA transfer. A software trigger transfer is initiated by writing a 1 to the DMAGO bit in CSR0. After the CSR0 write, the VBD requests the bus, performs the

DMA transfer specified by information in the control memory, and releases the bus. Hardware trigger is implemented with the SLVRDY* and DONE* control lines on the VME P3 connector. The user device asserting SLVRDY* begins a hardware trigger transfer. The VBD then requests the bus, performs the DMA transfer specified by information in the control memory, releases the bus, and then asserts DONE* until SLVRDY* is de-asserted by the user device. When SLVRDY* is de-asserted, DONE* is de-asserted.

Arbitration:

- arbitration is done only between block transfers, independent of the length of the block (<65535)
- after every block transfer, re-arbitrate.

⇒ No arbitration while a block transfer is in progress.

VMETRO Dump of the VBD parameter setting and DMA transfer.

As soon as we have our PCIH back from ZRL I will run an example for CAL crate environment.

Issues to be resolved as I see them:

Please send me your comments and instructions what and where to fix or add a specific detail. I mean don't just say add this material, I would like to also receive where to add the info, since as you can see from the manual things tend to repeat, as it should be for people like Gordon and I who just seek out and read only the relevant sections.

Ok, now we have something to work on, but it is only a Draft 0.1 so do not circulate it.
I apologize if the format is not up to standards; this is my first Word document.

VBD physical design and operation

Board designation.

VBD = VME **B**uffer **D**river

Function.

This board will serve as the read out device for the VME based front-end electronics in the DØ experiment, and output the collected data on a data cable to the Level-3 farm.

Design Goals.

- Generic design for all variations of front-end crates.
- Programmable VME interface controller.
- 30 Mbytes/sec VME transfer rate (20 Mbytes/sec min.).
- Rapid error recovery.
- On-board diagnostic/remote diagnostic
- Token ring readout arbitration

General Design.

The front-end electronic cards are housed in VME crates, with VME standard J1 and J2 back-planes. The VBD complies with the VME form factor, and is capable of acting as a DMA master on the VME bus. Each front-end VME crate contains a VBD. The board has two memory buffers, each of which can be placed either on the VME bus (and look like standard VME slave memory), or an external I/O port, which is connected to a data cable.
DMA transfers are used to move data from the front-end electronics cards into one of the buffers. Buffer access is then switched to the external port, and the data is put onto the data cable, under the control of the sequencer (or software control if VBD is in one of the diagnostic modes). The VME port and external data cable port, are each under the control of independent programmable controller to allow maximum flexibility of operation.

General Architecture.

The complexity and programmable nature of the VBD makes a coherent specification somewhat difficult to write. The architecture description has therefore been divided into eight functional/physical elements, which are described in the following pages.

This document contains updated information from the original memos and provides further explanations from the experience gained by Brown University L3 group, as well as, other groups and ZRL.

VBD Block Diagram

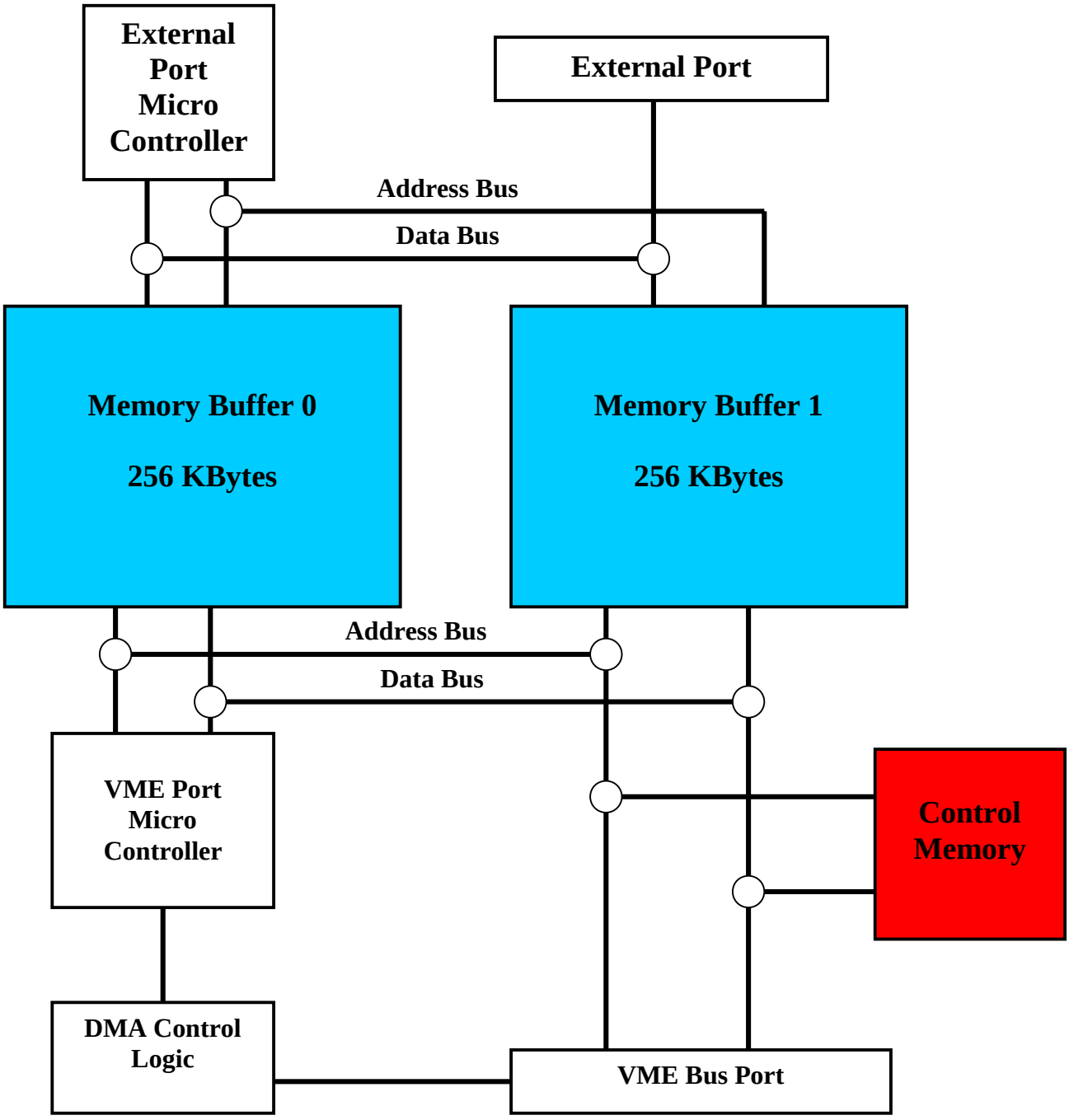

Figure 1

I Buffer Memory.

1. *Memory size*: 512 Kbytes.
2. *Configuration*: Dual 256 Kbytes buffers, each buffer has an associated independent I/O port.
3. *Timing:* Maximum cycle time of 90 *nsec* (including decode).
4. *Physical:* Four half-populated high-density SIP modules.
5. *Expansion:* On-board logic is capable of supporting additional memory; the existing SIPs can be replaced with fully populated modules to a limit of 1 Mbytes.

II VME Port.

- ***DMA master*:**
  Under normal operation, block mode transfer will be used to move data from the individual front-end electronic boards to the VBD on-board buffer memory. This is accomplished via an intelligent on-board list processing DMA engine, incorporated as part of the VME port controller.

  Two pieces of information are required for the transfer to take place, the starting *address(es)* of the data blocks and the *address(es)* of the associated word counts. Two different general modes of operation are present (programmed into VBD) (i.e. *Calorimeter and Central Detector modes*).

  In all cases the list of read-out locations (starting address) may be unique to the crate, and are stored in an on-board control memory. The control memory can be preprogrammed, i.e. loaded via the VME bus. The word counts, on the other hand, are unique to the event and must be read from memory locations associated with the front-end electronics. The DMA controller will assemble a list of word counts associated with its programmed read-out locations prior to the start of the transfer. A list of up to 512 word-count locations and the associated starting addresses of the data blocks can be stored in the VBD control memory.

  In the case of the *Calorimeter mode*, the word counts (i.e. long words = 32 bits) are located in I/O space, and the read-out locations reference the starting addresses for banks of memory which contain the data. The DMA controller will auto-increment the address pointer for successive reads of the data. The calorimeter electronics are not capable of supporting block mode transfers, however, they do support address pipelining. The calorimeter data does not include word counts, therefore the word count value is placed in the buffer by the VME controller prior to the start of the corresponding board transfer. Zero word count values are still posted to the data buffer, even though no data block transfer is performed.

  In the case of the *Central Detector* (CD), the word counts are located in I/O space, and the readout locations reference FIFO buffers which are filled to the specified word count. The DMA controller will issue that number of successive reads to the FIFO buffer address, without incrementing the address. The FADC boards are capable of performing as block mode slaves for a minimum of 256 consecutive transfers. The word counts are imbedded in the data, therefore, the controller does not add them.

- ***VME Slave Memory:***
  Each of the buffers can have its port switched to the VME bus, and appear as transparent VME memory. Under normal operation, one or both buffers may be ported to the VME. Buffer status is indicated in the CSR. The memory will support 8/16/32 bit-aligned transfers, with a cycle time of approximately 200 *nsec*. The default size of the memory buffers is 256 Kbytes for a total of 512 Kbytes, however, logic has been provided such that this can be doubled by use of denser SIP memories.

- ***Mailbox: (This is no longer true!!! What is the new format for the Mailbox)***
  The last 8 words of each buffer are used as a small scratchpad memory. Half of this memory (4 word) is reserved to assemble a trailer packet to accompany the data, which includes total word count, event number/crate id, token value and the checksum. The second half can be used to implement mailbox

registers to facilitate communication between the input and output port controllers and/or a VME bus device. These memory locations can be appended to the data stream by the controller. The choice of appending the data is determined initially by an on-board jumper (power-up default), and can be altered by token or CSR commands.

- ***Control Memory:***
  The base address values for DMA transfers and the word count locations are stored in the control memory, which is an independent memory region located in I/O space. This is a battery backed SRAM (self-contained) which can be programmed via the VME bus, or by an external programmer and then loaded into the board. The total *control memory* space is 16 Kbytes, however, only 8 Kbytes are available for external access from the VME bus. This region of memory is also used for the error status FIFO block (512 bytes) and the four CSRs. The remaining 4 Kbytes are available for data base addresses and the word/byte count locations, allowing lists up to 512 elements to be accommodated. Only word addressing (A16:D16) of the control memory is supported (note that 32 bit addresses must be programmed with 2 word transfers. The second 8 Kbytes section is reserved for the use by the VME I/O controller.

- ***Bus arbitration:***
  Standard VME signal level (SGL) bus request logic is implemented. The VBD will **not** release the bus until completion of transfer (RWD), unless ACFAIL, SYSRESET, or BCLR is detected (BCLR is only recognized between transfer descriptor element processing).

- ***Timing:***
  Timing diagrams for both block mode and non-block mode DMA, as well as, acquisition and arbitration timing can be found in the original VBD documentation.

III Data Cable Port.

The data cable port conforms to Futurebus electrical specifications, and follows the protocol and timing defined in the data cable specification. The VBD is responsible for the local generation of SYNC. The default data rate is 48 Mbytes/sec, however a slower rate may be selected by setting the appropriate CSR bit, or installing a jumper.

Data cable readout, event synchronization, diagnostic and initialization functions are all carried out under normal operation via a token passing scheme. This implementation is a mastered parallel token ring with multiple pass execution and variable token type. Programming and control of the external port is primarily carried out using tokens, and is described in more detail later.

IV Synchronization/request logic.

1. ***VME:*** An acquisition cycle begins when a slave request is signaled (SRDT), and VBD arbitrates for the bus. Only one slave request can be active at a time, i.e. there is no arbitration logic for the slave requests. Once bus mastership is granted, it is **not** relinquished until the entire transfer of the event has been completed (within a give crate). When the transfer is finished, buffer access is removed from the VME bus, and the DONE is asserted until SRDY is dropped. The SRDY and DONE signals are available on J3 C8 and J3 C10 respectively. They are both assumed to be open collector, negative true logic. On board pull-up can be provided if these lines are not terminated. They are assumed to meet minimum VME timing specifications.

2. ***Data cable:*** Upon completion of DMA transactions, buffer access and control are switched to the output I/O controller. The controller will wait for the appropriate readout token, at which time it will dump the buffer to the data cable. Once the trailer has been placed on the cable, the controller will pass the token to the next board. No physical interlocking of the buffers is done, however, the data cable protocol requires FIFO ordering of the event readout. If at the end of (x) μsec the buffer has not been read out, an error condition can be signaled.

V Diagnostics/power sequencing.

1. ***Diagnostics:*** All non-readout modes are classed as diagnostic functions. These functions can be used to do configuration programming, as well as VBD, data cable, and VME diagnostics. Diagnostic modes can be selected either by passing a diagnostic token from the sequencer/supervisor, by setting the appropriate bits in the CSR via the VME bus interface, or by switches. Some functions may also require programming the control memory, or passing parameters and function codes via mailbox registers. The most important diagnostic functions are as follows (possible access modes are given in parenthesis):

    - ***Bypass mode*** (*token/CSR/switch/jumper*): When logical bypass is selected, the board will ignore all but targeted initialization tokens and simply pass token to the next node.
    - ***Grant mode*** *(CSR/token/jumper):* When no sequencer is available, readout of the buffer can be initiated via the READY and GRANT signals available on the diagnostic port. When a buffer is to be readout, READY goes high, and readout begins when the external GRANT signal is received. The 4 lowest order bits of the event number are also passed on the diagnostic port.
    - ***Loop mode*** *(CSR/token/jumper):* In this mode the controller will fill the buffer with a known test pattern and cause it to be readout. The readout can be continues or loop count (switch can only enable continuous), and can be triggered by token, GRANT signal, or VME. Readout rate can be controlled by CSR, token or jumper.
    - ***Local mode*** *(CSR/jumper):* For diagnostic checks of the VME bus (or for the simulation of the D0 readout for L3 simulation as is done at Brown University), the VBD can be switched into local mode, such that DMA transfers can be triggered by setting a bit in the VBD-CSR0. The data will be transferred into the selected buffer, and the VBD will return to the slave mode, allowing the data to be verified from the VME (or with appropriate bit set the data could be dumped onto the cable after the DMA to an L3 node and be examined there).

2. ***Power up:*** Upon power-up or reset, the VBD will perform self-test, including memory test, enable slave mode VME access, and wait for SRDY. (***Brown Observation***: If the VBD is in one of the diagnostic modes it will try to read the crate id on power-up or reset in addition to above sequence)

VI Error handling.

Error conditions: The possible error conditions are divided into their origin class as follows:

VME bus errors:
- Non-existent memory time out.
- Bus arbitration time-out.
- Block mode error.

(*NOTE: time-out values are programmable via CSR and/or jumpers*)

Input transfer error (not bus related):
- Invalid event number.
- SRDY time-out (no available buffer)
- Invalid buffer request
- Data overrun (transfer > 256 Kb)
- SRDY dropped before DONE reply

Internal errors:
- VME I/O controller time-out
- Output I/O controller time-out
- Invalid function code

Output transfer errors:
- event synchronization (stale data)
- event synchronization (non-FIFO order)
- invalid token

Error signaling: Errors are always signaled in two ways: as a bit set in the CSR, or a single bit set on the token passing cable. Error type/status (byte value) is provided through the CSR or by a diagnostic token. The token bit is cleared by the next event readout token, and the next VME transfer clears the CSR bit. Since VME error checking can be quite long, a FIFO ordered error status block is maintained in the control memory. Error status values are combined with 8 bits of their associated event number to form a word field. Up to 256 error values can be stored in the FIFO block. The action taken on the output port when an error occurs will depend upon the initialization programming established by the sequencer.

Data integrity: The output port generates byte parity checks as per the data cable specification. In addition, on-the-fly checksum generation can be done on the outgoing data, using a pair of fast ALU's. A 32 bit 2's complement checksum is created, with carry truncation. The checksum value is appended to the crate trailer as the last word of the data block (and includes ALL previous data in the block).

VII Packaging.

Physical:

VME form factor – 9U x 280mm x 0.55"

Summary of I/O's:

VME port: P1, P2, P3 connectors; J1, J2 standard back-plane connections, J3 custom (pin grid header for WW). Power and ground derived only from J1, J2.

Data cable port:

One 64 and one 26 pin connector

Token arbitration:

2 coax connectors

Diagnostic port:

One 26 pin connector.

- grant
- ready
- 4 low order bits of event number

Indicators:

LED's located on the edge of board provide diagnostic and status information. The indicator values are arrayed on the board (top to bottom) are:

GLED11: bit 3 of event number
GLED10: bit 2 of event number
GLED09: bit 1 of event number
GLED08: bit 0 of event number
RLED07: Token receive enabled
RLED06: Read-out in progress
RLED05: Buffer 1 status (ON=external port, OFF=VME)
RLED04: Buffer 0 status (ON=external port, OFF=VME)
RLED03: External port controller error (self test only)
RLED02: VME bus error during VBD operation
RLED01: VME controller/memory error (self test only)
RLED00: Power

NOTE: Upon power-up, the VBD executes an on-board self-test routine. If the self-test is passed only RLED7 and RLED0 are ON. While SYSRESET is asserted, all LED's should be ON (provides LED test).

VIII Data format.

Data Block Header:

The new 8-longword-header format is shown below:

Total Word Count	**Event # | Crate ID**	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Filed Moniker	**Field Description**
<Total Word Count>	Total word count of the data block, including the header and the trailer.
<Event # | Crate ID>	The high order 16 bits of this location are used to store "L3 Transfer Number" associated with the data being digitized in the front-end crates. The "L3 Transfer Number" (16 bits) is read from a preprogrammed in the control memory address. This value is assumed to be valid when DMA transfer is requested, and is placed in the event number register prior to the start of the data transfer. The low order 16 bits are used to store the Crate ID number, which is also read from a pre-assigned location in I/O space.
<reserved>	--
<reserved>	--
<reserved>	--
<reserved>	--
<reserved>	--
<reserved>	--

Data Payload:

Calorimeter mode: Word count is appended to the beginning of the corresponding data block from each bank of memory readout by the controller. Zero (∅) word count values are preserved.

WC0	Data0	**WC1**	Data1	**WC2**	Data2	**WC3**	Data3	**WC4**	Data4	…

Central Detector mode: Word count is embedded in the data stream from FIFO's.

Data0	**Data1**	**Data2**	**Data3**	**Data4**	**Data5**	**Data6**	**Data7**	**Data8**	**Data9**	…

Data Block Trailer:

The new 10-longword-trailer format is a copy of the header plus a token and checksum. It is shown below:

Total Word Count	**Event # | Crate ID**	Res.	Res.	Res.	Res.	Res.	Res.	Token	**NegativeChecksum**

Filed Moniker	Field Description
<Total Word Count>	Total word count of the data block, including the header and the trailer.
<Event # | Crate ID>	The high order 16 bits of this location are used to store "L3 Transfer Number" associated with the data being digitized in the front-end crates. The "L3 Transfer Number" (16 bits) is read from a preprogrammed in the control memory address. This value is assumed to be valid when DMA transfer is requested, and is placed in the event number register prior to the start of the data transfer. The low order 16 bits are used to store the Crate ID number, which is also read from a pre-assigned location in I/O space.
<reserved>	--
<reserved>	--
<reserved>	--
<reserved>	--
<reserved>	--
<reserved>	--
<Token Value>	Value of token which enabled readout
<NegativeChecksum>	32 bit checksum of all previous values in the data block, excluding this checksum.

VBD Installation Summary

Summary of requirements to operate the VBD in diagnostic mode:

(1) Configure the on-board jumpers to select the base address for the buffer and control memory.

(2) Insert the board into standard VME crate (WITH POWER OFF!!!). Note that P3/C8 and P3/C10 and connected to SRDY and DONE respectively.

(3) Connect data and control cables to J7 and J5 respectively. Pin 1 is indexed in the silk screen. Connect the diagnostic cable (used to provide READY and GRANT) to J2. READY is J2/1, GRANT is J2/25, and the 4 low order bits of the event are on J2/(3,5,7,9).

(4) Power on board and program the control memory (parameter values are described below)

(5) Initiate transfer either by asserting SRDY or setting DMAGO bit in CSR.

Programming/Configuration of the board:

The programming and configuration of the board is controlled in three ways:

- Jumper setting (base address, physical config., and power up state)
- VME port programming (CSR's and control memory)
- Token programming (via external port)

Jumpers:

There are a total of 13 jumper block locations on the board. Some of these are factory configured, while others are under user control.
Note: At this time the only user configurable jumpers are MJ1 and CJ1. These are the only shunt programmed header blocks on the board.

The jumper functions are summarized as follows:
MJx: (Buffer memory base address) Jumpers which select the base address and size of the buffer memory. The present boards are pre-configured for 64 Kbytes SIP's, and therefore only MJ1 must be set. The address correspondence is as follows:

MJ1 Pin	--	VME address
1-2		A31
3-4		A30
5-6		A29
7-8		A28
9-10		A27
11-12		A26
13-14		A25
15-16		A24
17-18		A23
19-20		A22
21-22		A21
23-24		A20
25-26		A19

If the jumper is installed the corresponding bit is set low for the address match. Using 64 Kb SIP's, VME A18 will select the buffer, and the lower address bits select the location within the buffer.
Note: Factory configuration is F00000

CJx: (CSR base address) This jumper selects an 8 Kbytes region of I/O space for the CSR and control memory. The address correspondence is as follows:

CJ1 Pin --	I/O region base address
1-2	A15
3-4	A14
5-6	A13

If the jumper is installed, the corresponding address is set high.
Note: Factory configuration is 00

PJx: (Controller configuration) These jumpers define the reset defaults for the VME and external port controllers.
Note: These jumpers are pre-configured, ***DO NOT CHANGE***.